

Translation of Reference 3 (JP-A-09-091303)

(54) Title of Invention: DATA MANAGEMENT DEVICE

(57) Abstract:

PROBLEM TO BE SOLVED: A subject of the present invention is to provide a data management device which decreases the amount of information stored in an index table, and can shorten the retrieval time of data.

SOLUTION: A data management device of the present invention is equipped with a data part which stores data and discrimination information, hash converting means which converts the discrimination information into a hash value, a 1st index table which consists of blocks which store hash values, discrimination information, and storage positions, a 2nd index table which contains the hash values used as indexes of respective blocks, and retrieval means which converts the discrimination information into the hash value, specifies the block containing the hash value with comparing the magnitude of hash value in the 2nd index table, retrieves the hash value contained in the 1st index table, specifies an object to be retrieved by contrasting the discrimination information belonging to this hash value, and retrieves corresponding data with the discrimination information concerned.

[Claim]

[Claim 1]

A data management device, comprising:

a data part which contains a plurality of data and discrimination information for retrieval corresponding to this data;

hash converting means which converts the discrimination information into a hash value shorter than bit length of the discrimination information according to a hash function;

a 1st index table which consists of a plurality of blocks which contains hash value groups in which the hash values converted by the hash converting means are compiled in a plurality of predetermined ranges, discrimination information corresponding to this hash value group, and storage positions of this discrimination information in the data part;

a 2nd index table which contains hash values contained in heads of respective blocks in the 1st index table, and storage positions of this hash value in the 1st index table; and

retrieval means which converts the discrimination information, inputted at the time of data retrieval, into a hash value, specifies a block in the 1st index table where the same hash value is contained by comparing the magnitude of this hash value with the magnitude of each hash value contained in the 2nd index table, retrieves the same hash value in the block, specifies the same discrimination information as an object to be retrieved by contrasting the discrimination information corresponding to the retrieved hash value with the discrimination information inputted for retrieval, and retrieves corresponding data from the data part with this discrimination information.

[Detailed Explanation of the Invention]

[0001]

[Technical Field of the Invention]

The present invention relates to a data management device. In further detail, it relates to a data management device managing data, contained in a data part, by using an index which contains discrimination information, corresponding to each data, and storage positions of data.

[0002]

[Prior Art]

In general, in order to manage a plurality of data contained in memory means, retrieval efficiency is improved by making an index which contains discrimination information for retrieval, which is extracted from this data, and storage positions of data to shorten the access time to data.

[0003]

For example, as shown in Figure 5, a conventional data management device comprises a data part 50 which is formed in memory means 20, and contains a plurality of data and discrimination information which corresponds to this data, a first index table 40 which consists of two blocks 41 and 42 which contain 2 sets of discrimination information of this data and pointers showing a storage position of each data in the data part, a 2nd index table 30 which contained discrimination information used as indexes of the 1st index table 40, and retrieval means 13 which retrieves the data in the data part 50 by using this 1st index table 40 and the 2nd index table 30.

[0004]

Then, names are contained in the data part 50 as data, and employee numbers are contained as discrimination information. In addition, larger values among the discrimination information contained in respective blocks 41 and 42 are contained in the 2nd index table 30.

[0005]

In such a data management device, the case that a name of Ueda and an employee number of 5787 are retrieved will be explained. First, the employee number of 5787 is inputted into the retrieval means 13. Then, in the 2nd index table 30, the retrieval means 13 contrasts the employee number of 5787 and employee numbers contained in the 2nd index table 30. Then, a block 41, where the employee number of 5787 is contained, in the 1st index table 40 is specified.

[0006]

Next, in the block 41, the inputted employee number of 5787 and the employee numbers contained are contrasted, and the employee number of 5787 which is an object to be retrieved is specified. Finally, according to the designation of a pointer corresponding to this employee number of 5787, the name of Ueda is retrieved from the data part 50.

[0007]

[Problems to be Solved by the Invention]

However, like the above-mentioned example, the conventional data management device contains discrimination information in the first index table 40 and 2nd index table 30 to use it as a key at the time of data retrieval. For this reason, there is a problem that, in the case that the bit length of discrimination information is long or in the case of an index (compound index) for a plurality of discrimination

information, the amount of information contained in respective index tables 30 and 40 becomes large, and hence, access of data takes time.

[0008]

The present invention is made in consideration of the above-mentioned problems, and it is a technical issue to provide a data management device which decreases the amount of information stored in an index table, and can shorten the retrieval time of data.

[0009]

[Means for Solving the Problems]

The present invention is a data management device, and the following means are adopted in order to solve the above-mentioned technical subject.

[0010]

That is, it comprises a data part which contains a plurality of data and discrimination information for retrieval corresponding to this data, hash converting means which converts the discrimination information into a hash value shorter than the bit length of the discrimination information according to a hash function, a 1st index table which consists of a plurality of blocks which contains hash value groups in which the hash values converted by the hash converting means are compiled in a plurality of predetermined ranges, discrimination information corresponding to these hash value groups, and storage positions of this discrimination information in the data part, a 2nd index table which contains hash values contained in heads of respective blocks in the 1st index table, and storage positions of these hash values in the 1st index table, retrieval means which converts the discrimination information, inputted at the time of data retrieval, into a hash value, specifies a block where the same hash value is

contained by comparing this hash value with each hash value contained in the 2nd index table, retrieves the same hash value in the block, specifies the same discrimination information as an object to be retrieved by contrasting the discrimination information corresponding to the retrieved hash value and the discrimination information inputted for retrieval, and retrieves corresponding data from the data part with this discrimination information (in correspondence to claim 1).

[0011]

Hereafter, components indispensable to the present invention will be explained.

(Data part)

A data part contains a plurality of data and discrimination information for retrieval which corresponds to this data. The discrimination information may be what a part of data is specified, or may be what is specified separately. As the discrimination information, for example, when data is a name, then, age, an employee number, a date of birth, etc. are used. As for this data part, it is preferable to make it, for example, a table form, and to contain each data and discrimination information as a set.

[0012]

(Hash converting means)

Hash converting means lessens the amount of information contained in an index table by converting discrimination information into a hash value shorter than the bit length of the discrimination information according to a hash function. This hash converting means contains the hash function, and when discrimination information is inputted, it converts the discrimination information into a hash value. It is

also sufficient that the hash function just converts discrimination information into a short hash value on the basis of a fixed rule.
[0013]

For example, it is also good to define the hash function as follows.

Data = abcd

Hash function $h = (ab + cd) \bmod 100$

For example, with converting the discrimination information of 5443 by such a hash function, a hash value becomes 97, which becomes shorter than the discrimination information. This converted hash value is contained in the 1st index table and 2nd index table, and the retrieval of data contained in the data part is performed by retrieving this contained hash value.

[0014]

(1st index table)

The 1st index table becomes an index of each data contained in the data part. This 1st index table consists of a plurality of blocks. A hash value group where a plurality of hash values converted by the hash converting means is compiled in a predetermined range is contained in each block.

[0015]

Although this predetermined range can be determined freely, it is compiled so that each hash value group may have the almost same number of converted hash values in consideration of the distribution status of the converted hash values. In addition, the number of blocks is provided with corresponding to the number of these hash value groups. At this time, it is necessary not to take time for the retrieval in a block because the number of the hash values contained in the block

increases. Furthermore, it is preferable to arrange and contain the hash values, contained in each block, in descending or ascending order, and to contain a maximum value or a minimum value in each hash value group in the head of each block.

[0016]

For example, when four hash values of 99, 44, 91, and 97 exist, the hash values are lined up in descending order, one hash value group is formed by 99 and 97, and the predetermined range is determined within the hash values 99 to 92. Moreover, similarly, another hash value group is formed by 91 and 44, and the predetermined range is determined from 91 to 00. Then, the 1st index table with two blocks is formed, the hash value 99 is contained in one block as a head, and the hash value 91 is contained in another block as a head. In addition, after this, when the discrimination information of data added is converted into a hash value 70, the hash value 70 is contained in another block. But, when the number of the converted hash values increases with the increase in data, the predetermined range may be changed for a block count to be increased.

[0017]

Furthermore, the discrimination information corresponding to hash values is contained in each block. This, for example, when two pieces of discrimination information is converted into hash values, the hash values may become the same value (collision of a hash value) although values of the discrimination information differ. Accordingly, if this discrimination information is not contained, when data is retrieved by the hash value, it is not possible to judge which discrimination information the hash value corresponds to. Hence, there is a possibility that data retrieval may become inaccurate.

For this reason, this is intended to avoid a data retrieval error by the collision of a hash value by containing with the hash value the discrimination information which is an origin at the time of conversion into the hash value. Accordingly, when there is two or more discrimination information corresponding to one hash value, all of them are contained.

[0018]

Furthermore, storage positions of the discrimination information in the data part are contained in each block. It is good to contain these hash value, discrimination information, and storage position in each block as a set.

[0019]

In addition, it is also good to further contain the number of storage of the discrimination information belonging to one hash value in each block.

(2nd index table)

The 2nd index table achieves the duty of an index of the 1st index table. This 2nd index table contains hash values contained in heads of respective blocks of the 1st index table. Hash values contained in this 2nd index table are made indexes of respective blocks of the 1st index table. Furthermore, a storage position of this hash value, used as an index, in the 1st index table is contained. Moreover, it is preferable to contain the hash values, contained, with lining up in descending or ascending order.

[0020]

This 2nd index table makes it possible to retrieve the 2nd index table first on the occasion of data retrieval to specify in which block

of the 1st table a hash value into which the discrimination information of retrieval data is converted is contained.

[0021]

For example, in the case that upper and lower limits of hash values are 99 and 00 respectively, when the 1st index table is formed with a block whose predetermined range is within hash values from 0 to 50, and a block whose predetermined range is within hash values from 51 to 99, a hash value of 99 and a hash value of 50 are contained in the 2nd index table. Then, when a hash value of retrieval data is, for example, 70, then, 50 and 70 are contrasted, further, 99 and 70 are contrasted, it is specified that $50 < 70 \leq 99$, and a block whose predetermined range is from a hash value of 51 to a hash value of 99 is retrieved.

[0022]

In addition, the number of hash values contained in this 2nd index table corresponds to the number of blocks in the first index table. But, when this number of storage increases, it is also satisfactory not only to divide the hash values contained in a 2nd index into hash value groups with predetermined ranges, but also to make the 2nd index table consist of a plurality of blocks which contain each hash value group, and to make a 3rd index table in a further upper level to contain storage positions of the hash values contained in this 2nd index table, and hash values used as indexes of the hash values contained in the 2nd index table. In short, in the present invention, more than two levels of index tables can be formed according to the number of data contained, and any level can be also formed.

[0023]

(Retrieval means)

On the basis of a hash value converted by the hash converting means, retrieval means retrieves the 2nd index table and 1st index table in this order, and retrieves data contained in the data part.

[0024]

The retrieval means converts discrimination information, inputted at the time of data retrieval, into a hash value. Next, the magnitude of the converted hash value is compared with the magnitude of hash values contained in the 2nd index table. Then, a block where the same hash value is contained is specified.

[0025]

On the occasion of specification of this block, for example, predetermined ranges and values of heads of hash value groups are beforehand registered into the retrieval means. Then, a hash value group in which the converted hash value is included is specified by comparing the magnitude of the hash value, which the retrieval means converts, with the magnitude of each contained hash value in order. Then, a block of the 1st index table is retrieved according to the storage position of the 1st index table contained with corresponding to the hash value contained in the head of the specified hash value group.

[0026]

Next, the same hash value as the converted hash value is retrieved in the block containing the specified hash value group. This is performed, for example, by retrieving a matching hash value by sequentially contrasting the converted hash value with the hash values contained in the specific block.

[0027]

Next, the same discrimination information is retrieved by contrasting the discrimination information corresponding to the retrieved hash value and the discrimination information inputted for retrieval and this discrimination information is specified as an object to be retrieved. At this time, it is preferable at the point that the retrieval means can specify a frequency of contrasting discrimination information to contain the discrimination information corresponding to the hash value in the 1st index table.

[0028]

Finally, corresponding data is retrieved from the data part with this discrimination information. This is performed by retrieving the data part according to the storage position of the object data in the data part which is contained with corresponding to the discrimination information.

[0029]

(Appendant components of present invention)

Although the data management device of the present invention consists of indispensable components explained above, even if the appendant components explained below are added, it holds.

[0030]

(Registration means)

The data management device of the present invention may comprise registration means of registering new discrimination information and a storage position of data, corresponding to this discrimination information, in the data part as those belonging to a hash value into the 1st index table, when the hash value of new discrimination

information corresponding to the new data already exists in the 1st index table when the new data is added.

[0031]

(Deletion means)

In addition to the registration means, it may further comprise deletion means which retrieves a hash value of discrimination information corresponding to data to be deleted when the data is deleted, deletes the discrimination information concerned and hash value when one piece of discrimination information belonging to the hash value is contained in the 1st index table, and deletes the discrimination information corresponding to the data to be deleted when two or more pieces of discrimination information belonging to the hash value is contained.

[0032]

When it comprises registration means and deletion means in this way, the storage contents of the 1st index table and 2nd index table are changed with corresponding to the addition and deletion of data, and hence, it becomes possible that the retrieval means retrieves the 1st index table and 2nd index table, which have a suitable content, to perform data retrieval.

[0033]

(Action of present invention)

According to the data management device of the present invention, when data and discrimination information are contained in the data part, the hash converting means converts the discrimination information into a hash value, being shorter than the bit length of the discrimination information, according to a hash function. This hash value is contained in the 1st index table. Furthermore, a hash

value becoming a head of a hash value group is contained in the 2nd index table. In addition, discrimination information corresponding to the contained hash value is contained in the 1st index table, and a storage position of each data in the data part is further contained.

[0034]

Moreover, a storage position of the contained hash value in the 1st index table is contained in the 2nd index table, and a storage position of each hash value, which is contained in the 2nd index table, in the 1st index table is contained.

[0035]

Next, the action of the data management device at the time of data retrieval will be explained. When the discrimination information of data is inputted into the retrieval means, the retrieval means converts the inputted discrimination information into a hash value. A block in the 1st index table where the same hash value is contained by comparing the magnitude of this converted hash value with the magnitude of each hash value contained in the 2nd index table is specified in the 2nd index table.

[0036]

For example, when hash values of 99 and 91 are contained in the 2nd index table, suppose that the retrieval means converts certain discrimination information into a hash value of 93. In this case, the retrieval means compares the hash value of 93 with the hash value of 99 in the 2nd index table, and recognizes $93 \leq 99$. In addition, the retrieval means compares the hash value of 93 with the hash value of 91, and recognizes $91 < 93 \leq 99$. Thereby, a hash value group in which the converted hash value of 93 is included is specified. Thus, the hash value of 99 contained in the 2nd index table is recognized

to be an index of a block of the 1st index table whose predetermined range is from the hash value of 99 to the hash value of 92.

[0037]

In this way, when the block is specified, the retrieval means retrieves the same hash value as the converted hash value in the specified 1st index table by contrast them according to the designation of a position where the same hash value as the hash value contained in the 2nd index table is contained in the 1st index table. When this same hash value is retrieved, discrimination information which matches with the discrimination information contained with corresponding to this hash value by contrasting it with the hash value inputted for retrieval is specified as an object to be retrieved. When discrimination information is plural, this is repeatedly performed until the matching discrimination information is retrieved.

[0038]

Finally, corresponding data is retrieved from the data part according to the designation of a data storage position in the data part which corresponds to the discrimination information specified as the object to be retrieved. In addition, in the case that the number of storage of the discrimination information belonging to one hash value is contained in the 1st index table, when the number of storage of the discrimination information which belongs to the retrieved hash value is one when the same hash value is retrieved in the specific block of the 1st index table, the corresponding data is retrieved from the memory means by the discrimination information.

[0039]

On the other hand, when the number of storage of the discrimination information which belongs to the retrieved hash value

is plural, by contrasting the discrimination information inputted for retrieval with the plurality of discrimination information which belongs to the hash value, the same discrimination information is specified as an object to be retrieved. Then, the corresponding data is retrieved from the data part with this discrimination information.

[0040]

In addition, when the present invention is equipped with the registration means, it becomes as follows. When new data is added to the data part, discrimination information corresponding to the new data is converted into a hash value by the hash converting means. The registration means specifies a block, which contains the hash value converted in the 2nd index table, with this hash value, and investigates whether the same hash value has been already contained in the 1st index table. Then, when the hash value of new discrimination information has already existed in the 1st index table, the new discrimination information is registered into the 1st index table as what belongs to the hash value, and a storage position of the data in the data part which corresponds to this new discrimination information is registered further. On the other hand, when the hash value of the new discrimination information does not exist in the 1st index table, the hash value, discrimination information, and storage position in the data part are registered.

[0041]

Further, when the present invention is equipped with the deletion means, it becomes as follows. When deleting data from the data part, the deletion means retrieves a hash value of discrimination information corresponding to the data to be deleted in the 1st index table, similarly on the occasion of data retrieval. Then, when one

piece of discrimination information belonging to the hash value is contained, the discrimination information concerned, hash value, and storage position of the data to be deleted in the data part are deleted.

[0042]

On the other hand, when two or more pieces of discrimination information belonging to the hash value are contained, the deletion means deletes the discrimination information corresponding to the data to be deleted, and further deletes the storage position of the data to be deleted, corresponding to the discrimination information which is deleted, in the data part.

[0043]

[Embodiment of the Invention]

Hereafter, suitable embodiments of the data management device of the present invention will be explained in further detail by using drawings. Figure 1 shows the outline of a data management device 10 related to an embodiment of the present invention. This data management device 10 is equipped with an hash converting part 12, a retrieval part 13, a registration part 14, a deletion part 15, a 2nd index table 30, and a 1st index table 40.

[0044]

In addition, this data management device 10 contains and manages names which are data, and employee numbers which are discrimination information, in the data part 50 formed in a memory part 20.

<Hardware configuration of embodiment>

The data management device 10 according to an embodiment comprises in hardware a CPU, main memory, and auxiliary memory which are connected through a system bus. Among the components of the data management device 10, the hash converting part 12, retrieval part 13,

registration part 14, and deletion part 15 are programs contained in the auxiliary memory, and are functions which are reads into the main memory and are achieved by being loaded by the CPU.

[0045]

<Memory part>

The auxiliary memory is used as the memory part 20. A floppy disk, a hard disk, MO (Magnet Optical), etc. can be used for this auxiliary memory. The data part 50 for containing names and employee numbers with correspond to each other is formed in this memory part 20.

[0046]

<Data part>

The data part 50 is formed of two cell columns. Then, employee numbers are contained in a left-hand cell column respectively, and names corresponding to the employee numbers are contained in a right-hand cell column. Here, six persons' names are contained as data, and the employee numbers corresponding to respective person names are contained respectively.

[0047]

<Hash converting part>

The hash converting part 12 converts an inputted employee number into a hash value shorter than the bit length of the employee number according to a hash function. The hash function based on a fixed rule is contained in this hash converting part 12. A hash function used here is what partitions an employee number corresponding to a name every 1 byte, and totals respective partitioned values. When expressed in a formula,

Employee number = abcd

$$h = (ab + cd) \bmod 100$$

Accordingly, the upper limit of a hash value is 99 and a lower limit is 00.

[0048]

It is made that the hash converting part 12 sends this hash value to any one of the retrieval part 13, registration part 14, and deletion part 15 if needed.

<1st index table>

The 1st index table 40 is formed by a 1st block 41 and a 2nd block 42. Respective blocks 41 and 42 are formed of four cell columns. Hash values converted by the hash converting part 12 are arranged and contained in the descending order from a top in a left end cell column among this cell columns. In addition, numbers of employee numbers which belong to one hash value are contained in the 2nd cell column from the left. Furthermore, employee numbers which correspond to hash values are contained in the 3rd cell column from the left. Then, pointers which show storage positions of employee numbers and names in the data part 50 are contained in a right end cell column.

[0049]

Furthermore, when the collision of a hash value occurs, a cell, which contains the employee number, and a cell, which contains a pointer, are horizontally divided, and respective employee numbers are contained. Moreover, storage positions of employee numbers and names in the data part 50 are contained respectively in the cell which contains pointers.

[0050]

In a 1st block 41 of this 1st index table 40, it is made that its head is a hash value 99, that its predetermined range is from a

hash value of 99 to a hash value of 92, and that it contains hash values within this range. In addition, in a 2nd block 42, it is made that its head is a hash value of 91, that its predetermined range is from the hash value of 91 to a hash value of 00, and that it contains hash values within this range.

[0051]

<2nd index table>

The 2nd index table 30 is formed of two cell columns, and hash values contained in heads of respective blocks of the 1st index table 40 among hash values, converted by the hash converting part 12, are contained in the left-hand cell column in the descending order from a top. Pointers which show storage positions of respective hash values, contained here, in the 1st index table 40 are contained in the right-hand cell column.

[0052]

Here, the hash value of 99 and hash value of 91 are contained in the 2nd index table. This hash value of 99 becomes an index of the 1st block 41 containing a hash value group of the hash values from 99 to 92 in the 1st index table. Furthermore, the hash value of 91 becomes an index of the 2nd block 42 containing a hash value group of the hash values from 91 to 00.

[0053]

<Retrieval part>

The retrieval part 13 retrieves the 2nd index table 30 and 1st index table 40 in this order, and retrieves data from the data part 50.

[0054]

This retrieval part 13 has a hash function to be made to convert an employee number into a hash value when the employee number is inputted. The retrieval part 13 retrieves the 2nd index table 30 with the converted hash value to specify a block, where the same hash value as the converted hash value is contained, from one of the 1st block 41 and 2nd block 42. This is performed by specifying a hash value group, to which the hash value belongs, by comparing the magnitude of the converted hash value with the magnitude of the hash values contained in the 2nd index table 30 in order.

[0055]

The retrieval part 13 has the hash value of 99 and hash value of 91 which are head values of respective hash value groups, and retrieves the converted hash value in one of the 1st block 41 and 2nd block 42 of the 1st index table 40 according to the pointer contained with corresponding to the hash value 99 or hash value 91 contained in the head of the specified hash value group.

[0056]

Next, the same employee number is specified as an object to be retrieved by contrasting the employee number, corresponding to the retrieved hash value, with the employee number inputted for retrieval. Finally, according to the designation of a pointer corresponding to this employee number, a name is retrieved from the data part 50.

[0057]

<Registration part>

When a name and an employee number are inputted, it is made that the registration part 14 registers these in the data part 50. In addition, since the hash value is inputted from the hash converting

part 12, when the hash value is inputted, a block of the 1st index table which contains the hash value is specified from one of the 1st block 41 and 2nd block 42 in the 2nd index table 30. Next, in the specified block, the hash value, employee number corresponding to this hash value, and pointer which shows the storage position of this employee number in the data part 50 are contained, and one is further added to the number of storage of the employee number which belongs to one hash value.

[0058]

But in the specified block of the 1st index table 40, when the same hash value as the inputted hash value has been already contained, not only the employee number is registered in the 1st index table 40 as an object belonging to the hash value, but also one is added to the number of employees belonging to the hash value concerned.

[0059]

<Deletion part>

It is made that, when a name and an employee number are inputted, the deletion part 14 deletes the name and employee number from the data part 50. In addition, a hash value is inputted from the hash converting part 12. Then, first, in the 2nd index table 30, the magnitude of the inputted hash value and the contained hash value is compared, and a block where the same hash value is contained is specified from one of the 1st block 41 and 2nd block 42.

[0060]

Next, a hash value of the employee number corresponding to the deletion name is retrieved in the specified block of the 1st index table 40. Then, when the employee number belonging to the hash value is 1, not only the employee number, hash value, and pointer concerned

are deleted, but also one is subtracted from the number of storage of the employee number.

[0061]

At this time, when the number of storage of the employee number belonging to the hash value is plural, the employee number corresponding to the deleted name is specified with contrasting the inputted employee number with the employee numbers contained. Then, while deleting this employee number and pointer, one is subtracted from the number of the employee number which belongs to the hash value concerned.

[0062]

<Action of embodiment>

The action of the data management device 10 which is explained above will be explained by using Figures 1 and 2. For example, the operation of the data management 10 in the case of newly registering a name of Kagawa and an employee number of 5443 and the case of adding them will be explained.

[0063]

First, when the name of Kagawa and the employee number of 5443 are inputted into the registration part 14, the registration part 14 registers these in the data part 50. At this time, the employee number of 5443 is inputted into the hash converting part 12.

[0064]

Next, the hash converting part 12 converts the employee number of 5443 into a hash value 97. That is, $h = (54 + 43) \bmod 100 = 97$. This hash value of 97 is inputted into the registration part 14.

[0065]

Next, the registration part 14 compares the magnitude of the converted hash value of 97 with the magnitude of the hash value of 99 contained in the 2nd index table 30 to recognize $97 \leq 99$. Similarly, it compares the hash value of 97 with a hash value of 91 to recognize $91 < 97$. Thereby, the registration part 14 recognizes a predetermined range of the hash value group which contains the hash value of 97 to be $91 < 97 \leq 99$. At this time, the registration part 14 specifies from the head value of 99 of the hash value group, contained in the registration part 14, that the 1st block 41 where the hash value of 99 is contained in the head is a block where the hash value of 97 is contained.

[0066]

Next, the same hash value 97 is retrieved by contrasting the hash value of 97 converted in the 1st block 41, and the contained hash values sequentially from the head hash value of 99 according to the designation of the pointer corresponding to the hash value of 99 contained in the 2nd index table 30.

[0067]

At this time, suppose that the 1st block 41 does not contain the same hash value of 97. The registration part 14 registers the hash value of 97 in the 1st block 41 when recognizing that the hash value of 97 has not been yet contained. In addition, the employee number of 5443 which becomes the origin of being converted into this hash value of 97, and the pointer showing the storage position of the employee number of 5443 in the data part 50 are registered. Furthermore, one of the number of storage of the employee number which belongs to the hash value of 97 is converted.

[0068]

Moreover, when retrieving the same hash value of 97 in the case that the 1st block 41 has already contained the hash value of 97, the registration part 14 registers the employee number of 4354 in addition to the employee number of 5443 having already contained in the cell which contains the employee number corresponding to the hash value of 97. Furthermore, the pointer which shows the storage position of the employee number of 4354 in the data part 50 is registered. Furthermore, one is added to the number of storage, which is one, of the employee number which belongs to the hash value of 97 to be made two.

[0069]

Thus, when data is registered one after another, the data part 50 and 1st index table 40 become as shown in Figure 1. The operation of the data management device 10 in the case of retrieving, for example, a name of Sasaki and an employee number of 4354 from the data part 50 in such a data storage status will be explained below by using Figures 1 and 3.

[0070]

First, the employee number of 4354 is inputted into the retrieval means 13. Then, the retrieval part 13 calls the hash function and calculates $h = (43 + 54) \bmod 100 = 97$ to convert the employee number of 4354 into a hash value of 97. Next, the retrieval part 13 compares the converted hash value of 97 with the contained hash value of 99 in the 2nd index table 30 by using the hash value of 97, and recognizes $97 \leq 99$.

[0071]

Similarly, it compares the magnitude of the hash value of 97 and the hash value of 91, and recognizes $91 < 97$. Thereby, the retrieval part 13 recognizes that a predetermined range of the hash value group which contains the hash value of 97 is $91 < 97 \leq 99$. At this time, the retrieval part 13 specifies from the head value of 99 of the hash value group, contained in the retrieval part 13, that the 1st block 41 where the hash value of 99 is contained in the head is a block where the hash value of 97 is contained.

[0072]

Next, the same hash value 97 is retrieved by contrasting the hash value of 97 converted in the 1st block 41, and the contained hash values sequentially from the head hash value of 99 according to the designation of the pointer contained with corresponding to the hash value of 99 contained in the 2nd index table 30.

[0073]

When this same hash value 97 is retrieved, the retrieval part 13 next retrieves the number of storage of 2 of the employee number which belongs to this hash value of 97, and recognizes that what is necessary is just to perform the comparison of the employee numbers contained with the employee number of 4354 inputted for retrieval at a maximum of twice. Then, the same employee number of 4354 is retrieved in the 2nd contrast with contrasting the employee number sequentially from a top, and this same employee number of 4354 is specified as an object to be retrieved.

[0074]

Finally, the name of Sasaki is retrieved from the data part 50 according to the pointer which shows the storage position of the

employee number of 4354 in the data part 50 which is contained with corresponding to this employee number of 4354.

[0075]

When the number of storage of the employee number which belongs to the hash value of 97 is 1 at this time, the contrast of the employee number is performed only once, an object to be retrieved is specified, and the name of Sasaki is retrieved from the data part 50 according to the pointer corresponding to the employee number of 4354.

[0076]

Moreover, the retrieval part 13 sends a message of no applicable name when it is not possible to retrieve the hash value of 97 in the 1st index table 40 to make an output device such as a CRT display it.

[0077]

<Deletion of data>

Next, the operation of the data management device 10 in the case of deleting a name of Yamada and an employee number of 8110 which are contained in the data part 50 will be explained below by using Figures 1 and 4.

[0078]

First, when the name of Yamada and the employee number of 8110 are inputted into the deletion part 15, the employee number of 8110 is inputted into the hash converting part 12. Then, the hash converting part 12 calls the hash function and calculates $h = (81 + 10) \bmod 100 = 91$ to convert the employee number of 8110 into a hash value of 91. This hash value of 91 is sent to the deletion part 15.

[0079]

Next, the deletion part 15 compares the magnitude of this converted hash value of 91 with the magnitude of the hash value of

99 contained in the 2nd index table 30 to recognize $91 \leq 99$. Next, it compares the hash value of 91 with the hash value of 91 to recognize $91 \leq 91$. In this case, since the converted hash value of 91 coincides with the head value of 91 of the hash value group, the deletion part 15 specifies that the 2nd block 42 where the hash value of 91 is contained in the head is a block where the hash value of 91 is contained.

[0080]

Next, the same hash value 91 is retrieved by contrasting the hash value of 91 converted in the 1st block 41, with the contained hash values sequentially from the head hash value of 99 according to the designation of the pointer corresponding to the hash value of 91 contained in the 2nd index table 30.

[0081]

At this time, when the number of storage of the employee number belonging to the hash value of 91 is one, the employee number of 8110 corresponding to the deleted data is specified with contrasting once the inputted employee number of 8110 with the employee number of 8110 contained. Then, the hash value of 91, employee number of 8110, and pointer are deleted from the 1st index table 40. In addition, one is subtracted from the number of storage of the employee number which belongs to the hash value of 91.

[0082]

Then, the name of Yamada and the employee number of 8110 are deleted from the data part 50 on the basis of the pointer to be deleted. On the other hand, when the number of storage of an employee number is plural, that is, as shown in Figure 1, when an employee number of 1081 other than the employee number of 8110 is contained, it is recognized that what is necessary is just to perform the contrast of

the inputted employee number of 8110 with the employee numbers contained at a maximum of twice. Then, since the contrast is performed from a top in order of storing of employee numbers and employee numbers coincide in the first contrast, it is specified that this employee number of 8110 is the employee number of 8110 corresponding to the name to be deleted. Then, the employee number of 8110 and the pointer corresponding to this 8110 are deleted from the 2nd block 42 of the 1st index table 40. In addition, one is subtracted from the number of storage of the employee number which belongs to the hash value of 91.

[0083]

Then, the name of Yamada and the employee number of 8110 are deleted from the data part 50 on the basis of the pointer.

<Update of data>

The operation of the data management device in the case of updating a name contained in the data part 50 is performed similarly to the operation in the case of deleting a name and an employee number from the data part 50 and adding a name and an employee number subsequently. Hence, detailed explanation is omitted.

[0084]

<Effects of embodiment>

The effects of the data management device related to the embodiment explained above will be described below. According to the data management device 10, hash values converted by the hash converting means 12 are contained in the 2nd index table 30 and 1st index table 40. This hash value is a half in bit length in comparison with an employee number which is original discrimination information. Accordingly, the amount of information contained in the 2nd index table

30 and 1st index table 40 decreases as a whole. For this reason, the retrieval part 13 can perform data retrieval in a short time.

[0085]

In addition, it is possible to quickly retrieve the blocks 41 and 42, where the converted hash values are contained, because of the presence of the 2nd index table 30. Furthermore, it is possible to reorganize the index table 30 and 1st index table 40 which correspond to the newest storage content of the data part 50 because of the provision of the registration part 14 and deletion part 15. It is also possible to process this reorganization in a short time.

[0086]

In addition, since employee numbers are contained in the 1st index table 40, even if the collision of a hash value occurs, the retrieval means can retrieve data correctly.

[0087]

[Advantageous effects of the Invention]

According to the data management device of the present invention, since the discrimination information of data is converted into a hash value shorter than the bit length of the discrimination information according to a hash function by the hash converting means, it is possible to aim at decreasing the amount of information contained in index tables. For this reason, it is possible to shorten the retrieval time of data contained in the memory means. Furthermore, it is possible to adequately retrieve data in spite of occurrence of the collision of a hash value.

[Brief Description of the Drawings]

Figure 1 is a block diagram showing the outline of a data management device related to an embodiment of the present invention;

Figure 2 is a flowchart showing the procedure of operation of the data management device at the time of registering data in the data part;

Figure 3 is a flowchart showing the procedure of operation of the data management device at the time of retrieving the data contained in the data part;

Figure 4 is a flowchart showing the procedure of operation of the data management device at the time of deleting data from the data part; and

Figure 5 is a block diagram showing the outline of a conventional data management device..

[Description of Symbols]

12 ... Hash Converting Part

13 ... Retrieval Part

14 ... Register Part

15 ... Deletion Part

20 ... Memory Part

30 ... 2nd Index Table

40 ... 1st Index Table

41 ... 1st Block

42 ... 2nd Block

50 ... Data Part

【符号の説明】

【符号の説明】

20 記憶部

30 第2索引テーブル

40 第1索引テーブル

41 第1ブロック

42 第2ブロック

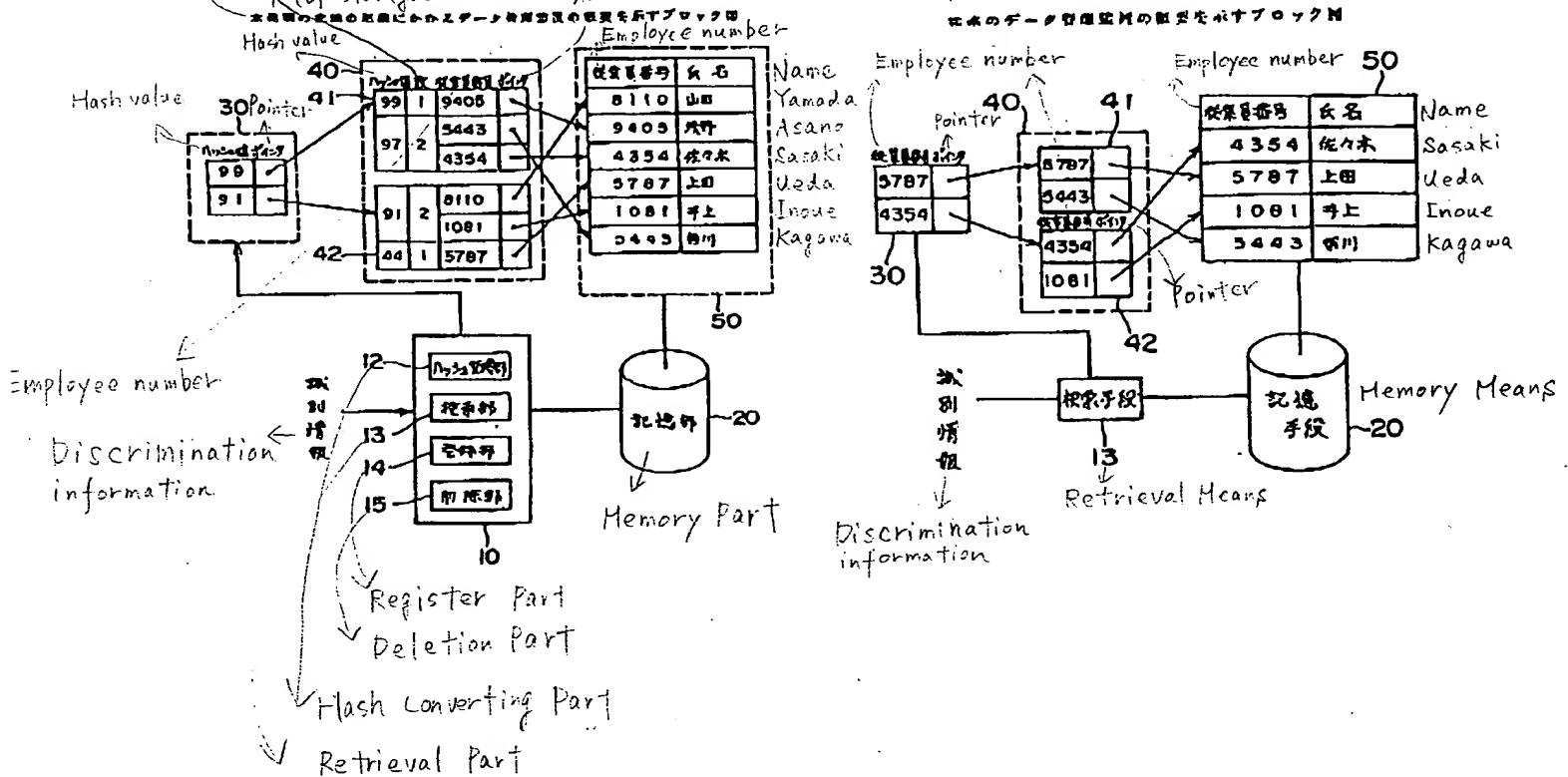
50 データ部

Block diagram showing the outline of a conventional data management device

Block diagram showing the outline of a data management device related to an embodiment of the present invention

【図5】
FIG.5

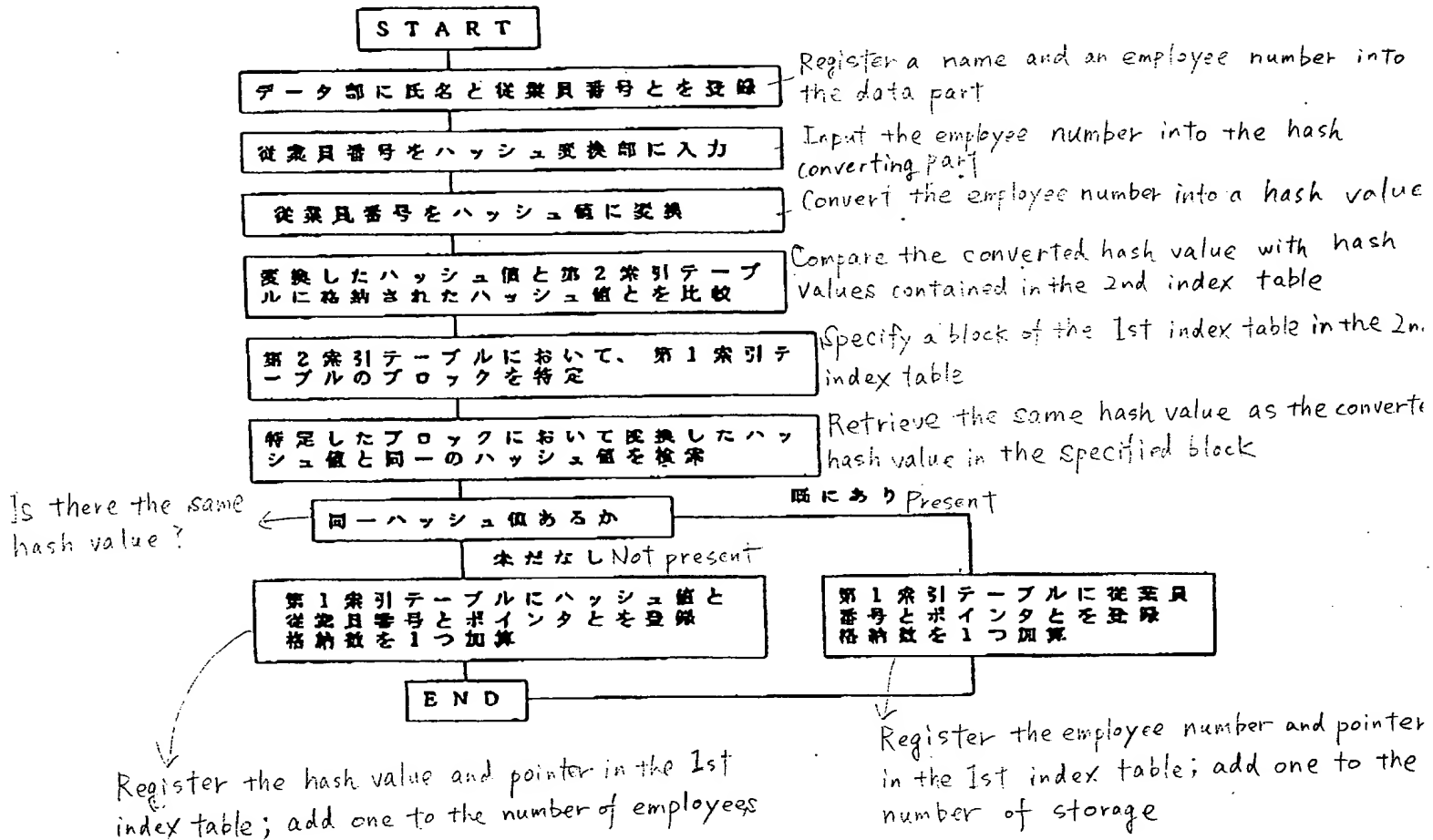
従来のデータ管理方式の形式を示すブロック図



Flowchart showing the procedure of operation of the data management device at the time of registering data in the data part (10) 特開平9-91303

【図2】FIG.2

データ部にデータを登録する際におけるデータ管理装置の動作の手順を示すフローチャート

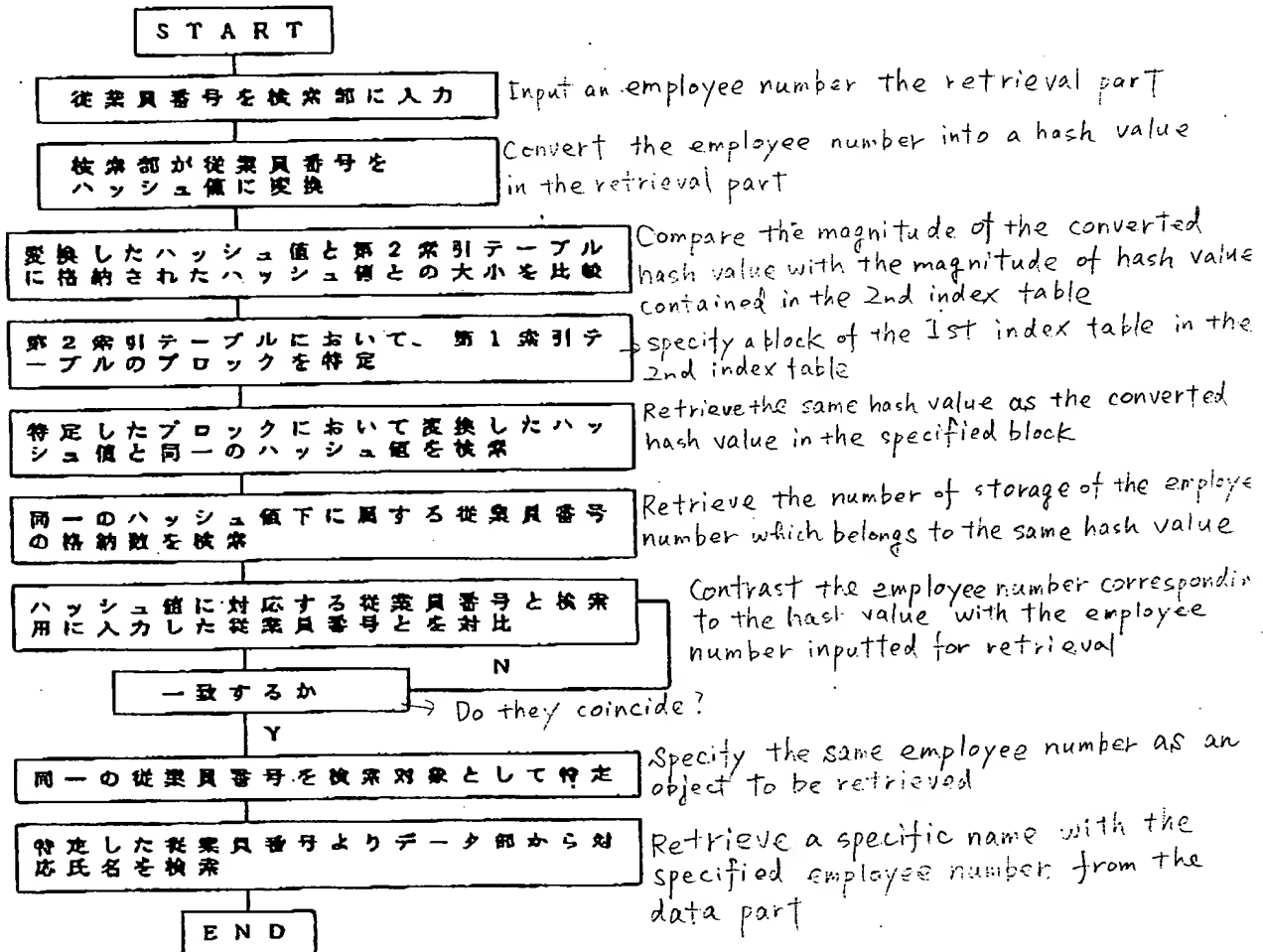


Flowchart showing the procedure of operation of the data management device at the time of retrieving the data contained in the data part

特開平9-91303

【図3】 FIG. 3

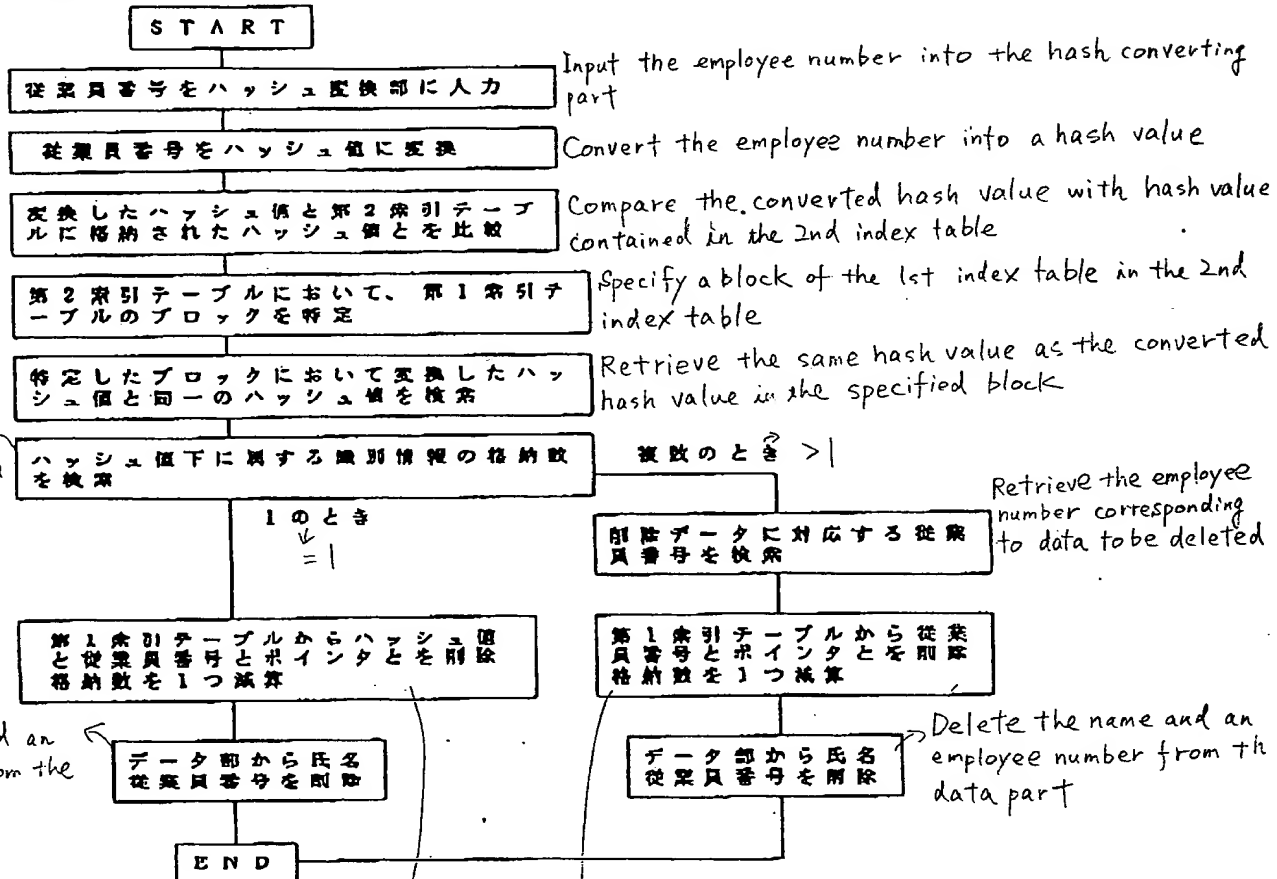
データ部に格納したデータを検索する際におけるデータ管理装置の動作の手順を示すフローチャート



Flowchart showing the procedure of operation of the data management device at the time of deleting data from the data part

【図4】FIG. 4

データ部からデータを削除する際におけるデータ管理装置の動作の手順を示すフローチャート



Retrieve the number of storage of the discrimination information which belongs the same hash value

Delete the name and an employee number from the data part

Retrieve the employee number corresponding to data to be deleted

Delete the name and an employee number from the data part

Delete the employee number and pointer from the 1st index table; subtract one from the number of storage

Delete the hash value, the employee number and pointer from the 1st index table; subtract one from the number of storage

フロントページの続き

(72)発明者 守屋 圭司
長野県松本市深志1丁目765番地 株式会社
社富士通パソコンラボ内

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-091303

(43)Date of publication of application : 04.04.1997

(51)Int.Cl. G06F 17/30
G06F 12/00

(21)Application number : 07-244861

(71)Applicant : FUJITSU LTD
PFU LTD

(22)Date of filing : 22.09.1995

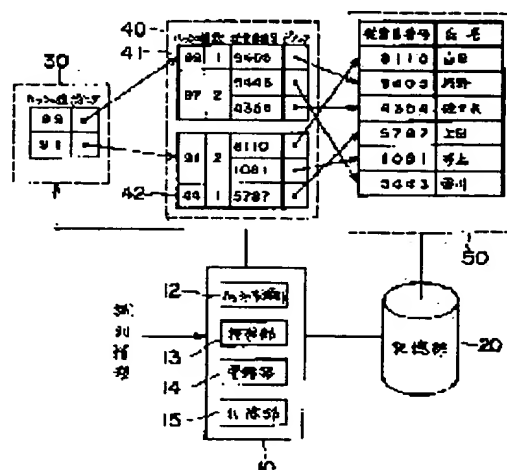
(72)Inventor : SHIMOYANAGI SATORU
MORIYA KEIJI

(54) DATA MANAGEMENT DEVICE

(57)Abstract:

PROBLEM TO BE SOLVED: To provide the data management device which can decrease the amount of information stored in an index table and shorten the retrieval time of data.

SOLUTION: This data management device is equipped with a data part 50 which stores data and discrimination information, a hash converting means 12 which converts the discrimination information into a hash value, a 1st index table 40 which consists of blocks containing hash values, discrimination information, and storage positions, a 2nd index table 30 which contain hash values as the indexes of the respective blocks, and a retrieval means 13 which converts the discrimination information into the hash value, specifies the block containing the hash value comparing the hash value in the 2nd index table 30, retrieves the hash value stored in the 1st index table 40, specifies an object to be retrieved by contrasting the discrimination information belonging to the hash value, and retrieves corresponding data with the discrimination information.



LEGAL STATUS

[Date of request for examination]

03.06.2002

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平9-91303

(43) 公開日 平成9年(1997)4月4日

(51) Int.Cl. ⁶	識別記号	庁内整理番号	FI	技術表示箇所
G 0 6 F 17/30			G 0 6 F 15/411	3 1 0
12/00	5 2 0		12/00	5 2 0 A

審査請求 未請求 請求項の数1 OL (全 12 頁)

(21) 出願番号 特願平7-244861

(22) 出願日 平成7年(1995)9月22日

(71) 出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中4丁目1番
1号

(71) 出願人 000136136

株式会社ピーエフユー

石川県河北郡宇ノ気町宇野気ヌ98番地の
2

(72) 発明者 下柳 悟

石川県河北郡宇ノ気町宇野気ヌ98番地の
2 株式会社ピーエフユー内

(74) 代理人 弁理士 遠山 勉 (外1名)

最終頁に続く

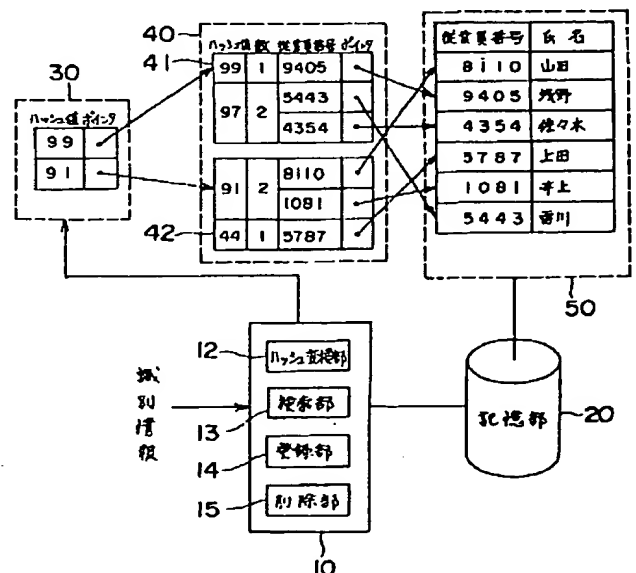
(54) 【発明の名称】 データ管理装置

(57) 【要約】

【課題】 本発明の課題は、索引テーブルに格納される情報量を減少し、データの検索時間を短縮しうるデータ管理装置を提供することである。

【解決手段】 本発明のデータ管理装置は、データと識別情報とを格納するデータ部と、識別情報をハッシュ値に変換するハッシュ変換手段と、ハッシュ値と識別情報と格納位置とを格納するブロックからなる第1索引テーブルと、各ブロックの見出しとなるハッシュ値を格納する第2索引テーブルと、識別情報をハッシュ値に変換し、第2索引テーブルでハッシュ値の大小を比較してハッシュ値を格納するブロックを特定し、第1索引テーブルで格納したハッシュ値を検索し、このハッシュ値に属する識別情報を対比して検索対象を特定し、当該識別情報により対応データを検索する検索手段とを備える。

本発明の実施の形態にかかるデータ管理装置の概要を示すブロック図



【特許請求の範囲】

【請求項 1】 複数のデータとこのデータに対応する検索用の識別情報とを格納するデータ部と、
前記識別情報をハッシュ関数に従って識別情報のビット長より短いハッシュ値に変換するハッシュ変換手段と、
前記ハッシュ変換手段により変換されたハッシュ値を複数所定範囲でまとめたハッシュ値群、このハッシュ値群に対応する識別情報、及び、この識別情報のデータ部における格納位置、を格納する複数のブロックからなる第 1 索引テーブルと、
前記第 1 索引テーブルの各ブロックの先頭に格納されるハッシュ値、及び、このハッシュ値の第 1 索引テーブルにおける格納位置、を格納する第 2 索引テーブルと、
データ検索時に入力された識別情報をハッシュ値に変換し、このハッシュ値と第 2 索引テーブルに格納された各ハッシュ値との大小を比較して同一のハッシュ値が格納された第 1 索引テーブルのブロックを特定し、そのブロックにおいて同一のハッシュ値を検索し、検索したハッシュ値に対応する識別情報と検索用に入力した識別情報とを対比して同一の識別情報を検索対象として特定し、この識別情報より前記データ部から対応データを検索する検索手段と、
を備えたデータ管理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明はデータ管理装置に関する。さらに詳細には、データ部に格納したデータを、各データに対応する識別情報とデータの格納位置とを格納した索引を用いて管理するデータ管理装置に関する。

【0002】

【従来の技術】 一般に、記憶手段に格納した複数のデータを管理するため、このデータから抽出した検索用の識別情報とデータの格納位置とを格納する索引を作成し、データへのアクセス時間を短縮して検索効率を向上するようにしている。

【0003】 例えば、図 5 に示すように、従来のデータ管理装置は、記憶手段 20 に形成され複数のデータと各データに対応する識別情報とを格納したデータ部 50 と、この各データの識別情報とデータ部における各データの格納位置を示すポイントとを二組格納する二つのブロック 41、42 からなる第 1 の索引テーブル 40 と、第 1 の索引テーブル 40 の見出しとなる識別情報を格納した第 2 の索引テーブル 30 と、この第 1 の索引テーブル 40 と第 2 の索引テーブル 30 とを用いてデータ部 50 のデータを検索する検索手段 13 とを備えている。

【0004】 そして、データ部 50 にはデータとして氏名が格納され、識別情報として従業員番号が格納されている。また、第 2 の索引テーブル 30 は、各ブロック 41、42 に格納した識別情報のうち大きい値が格納されている。

【0005】 このようなデータ管理装置において、例えば、氏名上田、従業員番号 5787 を検索する場合を説明する。まず、検索手段 13 に従業員番号 5787 を入力する。すると、検索手段 13 が、第 2 の索引テーブル 30 において、従業員番号 5787 と第 2 の索引テーブル 30 に格納されている従業員番号とを対比する。そして、第 1 索引テーブル 40 のうち、従業員番号 5787 が格納されるブロック 41 を特定する。

【0006】 次に、ブロック 41 において、入力された従業員番号 5787 と格納されている従業員番号とを対比して検索対象の従業員番号 5787 を特定する。最後にこの従業員番号 5787 に対応するポイントの指示に従い、データ部 50 から氏名上田を検索する。

【0007】

【発明が解決しようとする課題】 しかし、従来のデータ管理装置は、前記例のように、第 1 の索引テーブル 40 と第 2 の索引テーブル 30 とに識別情報を格納しデータ検索の際のキーとしている。このため、識別情報のビット長が長い場合や複数の識別情報に対する索引（複合インデックス）の場合は、各索引テーブル 30、40 に格納される情報量が大きくなり、データのアクセスに時間を要するという問題点がある。

【0008】 本発明は、前記問題点に鑑みなされたものであり、索引テーブルに格納される情報量を減少し、データへの検索時間を短縮し得るデータ管理装置を提供することを技術的課題とする。

【0009】

【課題を解決するための手段】 本発明は、データ管理装置であって、前記技術的課題を解決するために以下の手段を採用する。

【0010】 すなわち、複数のデータとこのデータに対応する検索用の識別情報とを格納するデータ部と、前記識別情報をハッシュ関数に従って識別情報のビット長より短いハッシュ値に変換するハッシュ変換手段と、前記ハッシュ変換手段により変換されたハッシュ値を複数所定範囲でまとめたハッシュ値群、このハッシュ値に対応する識別情報、及び、この識別情報のデータ部における格納位置、を格納する複数のブロックからなる第 1 索引テーブルと、前記第 1 索引テーブルの各ブロックの先頭に格納されるハッシュ値、及び、このハッシュ値の第 1 索引テーブルにおける格納位置、を格納する第 2 索引テーブルと、データ検索時に入力された識別情報をハッシュ値に変換し、このハッシュ値と第 2 の索引テーブルに格納された各ハッシュ値とを比較して同一のハッシュ値が格納されたブロックを特定し、そのブロックにおいて同一のハッシュ値を検索し、検索したハッシュ値に対応する識別情報と検索用に入力した識別情報とを対比して同一の識別情報を検索対象として特定し、この識別情報より前記データ部から対応データを検索する検索手段と、を備えている（請求項 1 に対応）。

【0011】以下、本発明に必須の構成要素を説明する。

(データ部) データ部は、複数のデータとこのデータに対応する検索用の識別情報を格納するものである。識別情報は、データの一部を指定したもので良く、また、別個に指定したもので良い。識別情報には、例えば、データが氏名である場合は、年齢、従業員番号、生年月日等が用いられる。このデータ部は、例えばテーブル状にし、各データと識別情報とを組として格納するのが好ましい。

【0012】(ハッシュ変換手段) ハッシュ変換手段は、識別情報をハッシュ関数に従って識別情報のビット長よりも短いハッシュ値に変換し、索引テーブルに格納する情報量を小さくするものである。このハッシュ変換手段は、ハッシュ関数を格納してあり、識別情報が入力されると、識別情報をハッシュ値に変換するようにしてある。ハッシュ関数は一定の法則に基づき識別情報を短いハッシュ値に変換できるものであれば良い。

【0013】例えば、ハッシュ関数を以下のように定義しても良い。

データ = a b c d

ハッシュ関数 $h = (a b + c d) \bmod 100$

このようなハッシュ関数により、例えば識別情報 5 4 4 3 を変換すると、ハッシュ値は 9 7 となり、識別情報よりも短いものとなる。この変換されたハッシュ値は、第 1 の索引テーブルと第 2 の索引テーブルとに格納され、この格納したハッシュ値を検索することによりデータ部に格納したデータの検索を行う。

【0014】(第 1 索引テーブル) 第 1 索引テーブルは、データ部に格納した各データの索引となるものである。この第 1 索引テーブルは複数のブロックからなる。各ブロックには、ハッシュ変換手段により変換されたハッシュ値を複数所定範囲でまとめたハッシュ値群が格納される。

【0015】この所定範囲は、自由に定め得るものであるが、変換されたハッシュ値の分布状況を考慮し、各ハッシュ値群がほぼ同数の変換されたハッシュ値を有するようにまとめる。なお、ブロックの数は、このハッシュ値群の数に対応して設ける。このとき、ブロックに格納するハッシュ値の数が多くなり、ブロック内の検索に時間がかからないようにする。また、各ブロックに格納されるハッシュ値は、大きい順または小さい順に並べて格納し、各ブロックの先頭には各ハッシュ値群の最大値または最小値を格納するのが好ましい。

【0016】例えば、ハッシュ値が 9 9, 4 4, 9 1 及び 9 7 の 4 つ存在する場合は、ハッシュ値を大きい順に並べ、9 9 と 9 7 とで一つのハッシュ値群を形成し、所定範囲をハッシュ値 9 9 から 9 2 までと定める。また、同様にして 9 1 と 4 4 とでもう一つのハッシュ値群を形成し、所定範囲を 9 1 から 0 0 と定める。そして、

第 1 索引テーブルを 2 つのブロックから構成し、ブロックの一方にハッシュ値 9 9 を先頭として格納し、ブロックの他方にハッシュ値 9 1 を先頭として格納する。なお、この後に、追加されたデータの識別情報がハッシュ値 7 0 に変換された場合は、ブロックの他方にハッシュ値 7 0 を格納する。もっとも、データの増加に伴い、変換されたハッシュ値の数が増加した場合は、所定範囲を変更してブロック数を増加しても良い。

【0017】また、各ブロックにはハッシュ値に対応する識別情報が格納される。これは、例えば二つの識別情報をハッシュ値に変換すると、識別情報の値が異なるにも拘わらずハッシュ値が同一値となる場合がある(ハッシュ値の衝突)。従って、この識別情報を格納しないと、ハッシュ値によりデータを検索した場合に、いずれの識別情報に対応するハッシュ値であるのか判断できず、データ検索が不正確なものとなるおそれがある。このため、ハッシュ値を変換する元となった識別情報をハッシュ値とともに格納することにより、ハッシュ値の衝突によるデータ検索ミスを回避しようとするものである。従って、一つのハッシュ値に対応する識別情報が複数ある場合は、その全てを格納する。

【0018】さらに、各ブロックには、データ部における識別情報の格納位置を格納する。これらのハッシュ値と識別情報と格納位置とは一組として各ブロックに格納するのが良い。

【0019】なお、各ブロックにはさらに一つのハッシュ値に属する識別情報の格納数を格納するようにしても良い。

(第 2 索引テーブル) 第 2 索引テーブルは、第 1 索引テーブルの索引の役目を果たすものである。この第 2 索引テーブルは、第 1 索引テーブルの各ブロックの先頭に格納されるハッシュ値を格納する。この第 2 索引テーブルに格納されるハッシュ値は、第 1 索引テーブルの各ブロックの見出しとされる。また、この見出しとなるハッシュ値の第 1 索引テーブルにおける格納位置を格納する。なお、格納されるハッシュ値は、大きい順から小さい順にならべて格納するのが好ましい。

【0020】この第 2 索引テーブルにより、データ検索の際に、まず第 2 索引テーブルを検索し、第 1 テーブルのどのブロックに検索データの識別情報を変換したハッシュ値が格納されているかを特定できるようにする。

【0021】例えば、ハッシュ値が上限を 9 9 とし下限を 0 0 とする場合において、0 から 5 0 までのハッシュ値を所定範囲とするブロックと、5 1 から 9 9 までのハッシュ値を所定範囲とするブロックとから第 1 索引テーブルが形成されている場合には、第 2 索引テーブルにハッシュ値 9 9 とハッシュ値 5 0 とを格納する。そして、検索データのハッシュ値が例えば 7 0 である場合は、5 0 と 7 0 を対比し、さらに 9 9 と 7 0 とを対比し、5 0 < 7 0 ≤ 9 9 と特定し、ハッシュ値 5 1 からハッシュ値

30

40

50

99までを所定範囲とするブロックを検索するようにする。

【0022】なお、この第2索引テーブルに格納するハッシュ値の数は、第1の索引テーブルのブロック数に対応するものであるが、この格納数が増加した場合は、第2の索引に格納されたハッシュ値を所定範囲のハッシュ値群に分けるとともに、第2索引テーブルを各ハッシュ値群を格納する複数のブロックからなるようにし、さらに上位に第3の索引テーブルを作成し、この第2の索引10 テーブルに格納したハッシュ値の格納位置と、第2の索引テーブルに格納したハッシュ値の見出しとなるハッシュ値とを格納するようにしても良い。要するに本発明は、格納されるデータの数に応じて索引テーブルを2階層以上に形成しても良いものであり、幾階層に形成しても良いものである。

【0023】（検索手段）検索手段は、ハッシュ変換手段により変換したハッシュ値に基づき、第2索引テーブル、第1索引テーブルの順に検索し、データ部に格納したデータを検索するものである。

【0024】検索手段は、まずデータ検索時に入力された識別情報をハッシュ値に変換する。次に、第2索引20 テーブルにおいて、変換されたハッシュ値と第2索引テーブルとに格納されたハッシュ値との大小を比較する。そして、同一のハッシュ値が格納されたブロックを特定する。

【0025】このブロックの特定にあたり、例えば、検索手段に、予めハッシュ値群の所定の範囲と先頭の値を登録しておく。そして、検索手段が変換したハッシュ値と格納された各ハッシュ値との大小を順に比較して、変換したハッシュ値が含まれるハッシュ値群を特定するよ30 うにする。そして、特定されたハッシュ値群の先頭に格納されるハッシュ値に対応して格納された第1索引テーブルの格納位置に従って第1索引テーブルのブロックを検索するようにする。

【0026】次に、特定したハッシュ値群を格納するブロックにおいて変換したハッシュ値と同一のハッシュ値を検索する。これは、例えば、変換したハッシュ値と特定ブロックに格納されたハッシュ値とを順次対比し、一致するハッシュ値を検索することにより行う。

【0027】次に検索した同一のハッシュ値に対応して格納されている識別情報と検索用に入力した識別情報とを対比して同一の識別情報を検索し、この識別情報を検索対象として特定する。このとき、第1索引テーブルにハッシュ値に対応する識別情報を格納しておく、検索手段が識別情報同士を対比する回数を指定することができる点で好ましい。

【0028】最後に、この識別情報より前記データ部から対応データを検索する。これは、識別情報に対応して格納されているデータ部における対象データの格納位置に従ってデータ部を検索することにより行う。

【0029】（本発明の付随的構成要素）本発明のデータ管理装置は、以上説明した必須の構成要素からなるものであるが、以下に説明する付随的構成要素を付加した場合であっても成立する。

【0030】（登録手段）本発明のデータ管理装置は、新規データを追加したとき、第1索引テーブルにおいて、前記新規データに対応する新規識別情報のハッシュ値がすでに存在するときは、そのハッシュ値下に属するものとして前記新規識別情報とこの識別情報に対応するデータのデータ部における格納位置とを前記第1索引10 テーブルに登録する登録手段を備えていても良い。

【0031】（削除手段）登録手段に加え、さらに、データを削除する際、削除データに対応する識別情報のハッシュ値を検索し、第1索引テーブルにおいて、前記ハッシュ値に属する識別情報が1つ格納されているとき、当該識別情報とハッシュ値とを削除し、そのハッシュ値に属する識別情報が複数格納されているとき、削除データ対応の識別情報を削除する削除手段を備えても良い。

【0032】このように登録手段と削除手段とを備えると、データの追加と削除とに対応して第1索引テーブルと第2索引テーブルとの格納内容が変更されることとなり、検索手段は、適切な内容の第1索引テーブルと第2索引テーブルとを検索してデータ検索を行うことが可能となる。

【0033】（本発明の作用）本発明のデータ管理装置によると、データと識別情報とがデータ部に格納されると、ハッシュ変換手段は、識別情報をハッシュ関数に従って識別情報のビット長より短いハッシュ値に変換する。このハッシュ値は、第1索引テーブルに格納される。また、ハッシュ値群の先頭となるハッシュ値は、第2索引テーブルに格納される。また、第1索引テーブルには、格納したハッシュ値に対応する識別情報が格納され、さらにデータ部における各データの格納位置が格納される。

【0034】また、第2索引テーブルには、格納したハッシュ値の第1の索引テーブルにおける格納位置が格納され、第2索引テーブルに格納した各ハッシュ値の第1索引テーブルにおける格納位置が格納される。

【0035】次に、データ検索時のデータ管理装置の作用を説明する。検索手段にデータの識別情報が入力されると、検索手段は、まず、入力された識別情報をハッシュ値に変換する。次に、第2索引テーブルにおいて、変換したハッシュ値と第2索引テーブルに格納されたハッシュ値とを比較して同一のハッシュ値が格納された第1索引テーブルのブロックを特定する。

【0036】例えば、第2索引テーブルにハッシュ値99と91が格納されている場合において、検索手段がある識別情報をハッシュ値93に変換したとする。この場合、検索手段は第2索引テーブルにおいて、ハッシュ値93とハッシュ値99とを比較し $93 \leq 99$ と認識す50

る。また、検索手段は、ハッシュ値 93 とハッシュ値 91 とを比較して $91 < 93 \leq 99$ と認識する。これにより、変換したハッシュ値 93 が含まれるハッシュ値群を特定する。すなわち、第 2 索引テーブルに格納されたハッシュ値 99 は、ハッシュ値 99 からハッシュ値 92 までを所定範囲とする第 1 検索テーブルのブロックの見出しと認識する。

【0037】このように、ブロックを特定すると、検索手段は、第 2 索引テーブルに格納されているハッシュ値と同一のハッシュ値が第 1 索引テーブルに格納されている位置の指示に従い、特定した第 1 索引テーブルにおいて、変換したハッシュ値と同一のハッシュ値を対比することにより検索する。この同一のハッシュ値を検索すると、このハッシュ値に対応して格納された識別情報と検索用に入力されたハッシュ値とを対比して一致する識別情報を検索対象として特定する。これは、識別情報が複数の場合は、一致する識別情報を検索するまで繰り返して行われる。

【0038】最後に、検索対象として特定した識別情報に対応するデータ部のデータ格納位置の指示に従いデータ部から対応データを検索する。なお、第 1 検索テーブルに一つのハッシュ値に属する識別情報の格納数が格納されている場合は、第 1 索引テーブルの特定ブロックにおいて同一のハッシュ値を検索した際、検索したハッシュ値下に属する識別情報の格納数が一つのときはその識別情報により記憶手段から対応データを検索する。

【0039】一方、検索したハッシュ値下に属する識別情報の格納数が複数のときは検索用に入力された識別情報とハッシュ値下に属する複数の識別情報とを対比して、同一の識別情報を検索対象として特定する。そして、当該識別情報によりデータ部から対応データを検索する。

【0040】また、本発明が登録手段を備える場合は以下になる。データ部に新規データを追加した際、新規データに対応する識別情報はハッシュ変換手段によりハッシュ値に変換される。登録手段は、このハッシュ値により、第 2 索引テーブルにおいて変換したハッシュ値を格納したブロックを特定し、第 1 索引テーブルにおいて既に同一のハッシュ値が格納されているかを調べる。そして、新規識別情報のハッシュ値が既に第 1 索引テーブルに存在するときは、新規識別情報をそのハッシュ値下に属するものとして第 1 索引テーブルに登録し、さらにこの新規識別情報に対応するデータのデータ部における格納位置を登録する。一方、新規識別情報のハッシュ値が第 1 索引テーブルに存在しない場合は、ハッシュ値、識別情報、データ部における格納位置を登録する。

【0041】さらに、本発明が削除手段を備える場合は以下になる。データ部からデータを削除する際、削除手段は削除データに対応する識別情報のハッシュ値

をデータ検索の際と同様にして、第 1 索引テーブルにおいて検索する。そして、そのハッシュ値に属する識別情報が 1 つ格納されているとき、当該識別情報とハッシュ値と削除データのデータ部における格納位置を削除する。

【0042】一方、そのハッシュ値に属する識別情報が複数格納されているとき、削除手段は削除データ対応の識別情報を削除し、さらに削除した識別情報に対応する削除データのデータ部における格納位置を削除する。

10 【0043】

【発明の実施の形態】以下、本発明のデータ管理装置の好適な実施の形態を図を用いてさらに詳細に説明する。図 1 には、本実施の形態にかかるデータ管理装置 10 の概要が示されている。このデータ管理装置 10 は、ハッシュ変換部 12 と、検索部 13 と、登録部 14 と、削除部 15 と、第 2 索引テーブル 30 と、第 1 索引テーブル 40 とを備える。

20 【0044】なお、このデータ管理装置 10 は、記憶部 20 に形成したデータ部 50 にデータである氏名と識別情報である従業員番号とを格納して管理するものである。

〈実施の形態のハード的構成〉実施の形態にかかるデータ管理装置 10 は、ハード的には、システムバスを介して接続された CPU と主記憶装置と補助記憶装置とから構成されている。データ管理装置 10 の構成要素のうち、ハッシュ変換部 12 と検索部 13 と登録部 14 と削除部 15 とは、補助記憶装置に格納されたプログラムであり、必要に応じて主記憶装置に読み出され、CPU によりロードされて実現する機能である。

30 【0045】〈記憶部〉記憶部 20 には補助記憶装置が用いられている。この補助記憶装置は、フロッピーディスク、ハードディスク、MO (Magnet Optical) 等を用いることができる。この記憶部 20 に氏名と従業員番号を対応させて格納するためのデータ部 50 が形成されている。

【0046】〈データ部〉データ部 50 は、二つのセル縦列により形成されている。そして、左側のセル縦列に従業員番号がそれぞれ格納され、右側のセル縦列に従業員番号に対応する氏名が格納される。ここでは、6 人の氏名がデータとして格納されており、また、各氏名に対応する従業員番号がそれぞれ格納されている。

【0047】〈ハッシュ変換部〉ハッシュ変換部 12 は、入力された従業員番号をハッシュ関数に従って従業員番号のビット長よりも短いハッシュ値に変換するものである。このハッシュ変換部 12 には一定の法則に基づくハッシュ関数が格納されている。ここで用いられているハッシュ関数は、氏名に対応する従業員番号を 1 バイト単位に区切り、区切ったそれぞれの値を合計するものである。数式で表せば、
従業員番号 = a b c d

$h = (ab + cd) \bmod 100$

である。従ってハッシュ値の上限は 99 であり、下限は 00 である。

【0048】ハッシュ変換部 12 は、このハッシュ値を必要に応じて検索部 13、登録部 14、削除部 15 のいずれかに送るようにしてある。

〈第 1 索引テーブル〉第 1 索引テーブル 40 は、第 1 ブロック 41 と第 2 ブロック 42 とから形成されている。各ブロック 41、42 は、4 つのセル縦列から形成されている。このセル縦列のうち、左端のセル縦列には、ハッシュ変換部 12 により変換されたハッシュ値が上から大きい順に並べて格納される。また、左から 2 番目のセル縦列は、一つのハッシュ値に属する従業員番号の数が格納される。さらに、左から 3 番目のセル縦列はハッシュ値に対応する従業員番号が格納される。そして、右端のセル縦列は、データ部 50 における従業員番号と氏名との格納位置を示すポインタが格納される。

【0049】なお、従業員番号を格納するセルとポインタを格納するセルとは、ハッシュ値の衝突が生じる場合には横に分割され、それぞれの従業員番号が格納される。また、ポインタを格納するセルには、データ部 50 における各従業員番号及び氏名の格納位置がそれぞれ格納される。

【0050】この第 1 索引テーブル 40 の第 1 ブロック 41 は、先頭をハッシュ値 99 とし所定範囲をハッシュ値 99 からハッシュ値 92 までとし、この範囲のハッシュ値を格納するものである。また、第 2 ブロック 42 は、先頭をハッシュ値 91 とし所定範囲をハッシュ値 91 からハッシュ値 00 までとし、この範囲のハッシュ値を格納するものである。

【0051】〈第 2 索引テーブル〉第 2 索引テーブル 30 は、二つのセル縦列から形成され、左側のセル縦列には、ハッシュ変換部 12 により変換されたハッシュ値のうち、第 1 索引テーブル 40 の各ブロックの先頭に格納されるハッシュ値が、上から大きい順に格納される。右側のセル縦列にはここに格納された各ハッシュ値の第 1 索引テーブル 40 における格納位置を示すポインタが格納される。

【0052】ここでは、第 2 索引テーブルには、ハッシュ値 99 とハッシュ値 91 とが格納されている。このハッシュ値 99 は、第 1 索引テーブルにおけるハッシュ値 99 から 92 までのハッシュ値群を格納する第 1 ブロック 41 の見出しとなっている。また、ハッシュ値 91 は、ハッシュ値 91 から 00 までのハッシュ値群を格納する第 2 ブロック 42 の見出しとなっている。

【0053】〈検索部〉検索部 13 は、第 2 索引テーブル 30、第 1 索引テーブル 40 の順に検索してデータ部 50 からデータを検索するものである。

【0054】この検索部 13 は、ハッシュ関数を用い、従業員番号が入力されると、従業員番号をハッシュ値に

変換するようにされている。検索部 13 は、変換したハッシュ値により第 2 索引テーブル 30 を検索し、変換したハッシュ値と同一のハッシュ値が格納されているブロックを、第 1 ブロック 41 と第 2 ブロック 42 とのいずれか一方から特定する。これは、変換したハッシュ値と、第 2 索引テーブル 30 に格納されているハッシュ値との大小を順に比較することにより、ハッシュ値が属するハッシュ値群を特定することにより行う。

【0055】検索部 13 は、各ハッシュ値群の先頭の値であるハッシュ値 99 とハッシュ値 91 とを有しており、その特定したハッシュ値群の先頭に格納されているハッシュ値 99 またはハッシュ値 91 に対応して格納されたポインタに従い、第 1 索引テーブル 40 の第 1 ブロック 41 と第 2 ブロック 42 との一方において変換したハッシュ値を検索する。

【0056】次に、検索したハッシュ値に対応する従業員番号と検索用に入力した従業員番号とを対比し、同一の従業員番号を検索対象として特定する。最後に、この従業員番号に対応して格納されたポインタの指示に従いデータ部 50 から氏名を検索するようにされている。

【0057】〈登録部〉登録部 14 は、氏名と従業員番号とが入力されると、データ部 50 にこれらを登録するようにされている。また、ハッシュ変換部 12 からハッシュ値が入力されるようになっており、ハッシュ値が入力されると、第 2 索引テーブル 30 において、ハッシュ値を格納する第 1 索引テーブルのブロックを第 1 ブロック 41 と第 2 ブロック 42 との一方から特定する。次に特定したブロックにおいて、ハッシュ値とこのハッシュ値に対応する従業員番号とこの従業員番号のデータ部 50 における格納位置を示すポインタとを格納し、さらに、一つのハッシュ値下に属する従業員番号の格納数を 1 つ加算するようにされている。

【0058】もっとも、第 1 索引テーブル 40 の特定したブロックにおいて、入力されたハッシュ値と同一のハッシュ値が既に格納されている場合は、そのハッシュ値に属するものとして従業員番号を第 1 索引テーブル 40 に登録するとともに、当該ハッシュ値に属する従業員番号数の値を 1 つ加算するようにされている。

【0059】〈削除部〉削除部 14 は、氏名と従業員番号とが入力されると、データ部 50 から氏名と従業員番号とを削除するようにされている。また、ハッシュ変換部 12 からハッシュ値が入力される。すると、まず、第 2 索引テーブル 30 において、入力されたハッシュ値と格納されたハッシュ値との大小を比較して、同一のハッシュ値が格納されたブロックを第 1 ブロック 41 と第 2 ブロック 42 との一方から特定する。

【0060】次に、特定した第 1 索引テーブル 40 のブロックにおいて、削除氏名に対応する従業員番号のハッシュ値を検索する。そして、そのハッシュ値に属する従業員番号が 1 のとき、当該従業員番号とハッシュ値とポ

インタとを削除するとともに従業員番号の格納数を1つ減算するようにされている。

【0061】このとき、ハッシュ値に属する従業員番号の格納数が複数のときは、入力された従業員番号と格納されている従業員番号とを対比して、削除氏名対応の従業員番号を特定する。そして、この従業員番号とポイントとを削除するとともに、当該ハッシュ値下に属する従業員番号の数の値を1つ減算するようにされている。

【0062】〈実施の形態の作用〉以上説明したデータ管理装置10の作用を図1と図2とを用いて説明する。例えば、氏名香川、従業員番号5443を新規に登録する場合と追加する場合におけるデータ管理10の動作について説明する。

【0063】まず、登録部14に氏名香川、従業員番号5443が入力されると、登録部14はこれらをデータ部50に登録する。このとき、ハッシュ変換部12に従業員番号5443が入力される。

【0064】次にハッシュ変換部12は従業員番号5443をハッシュ値97に変換する。すなわち $h = (54 + 43) \bmod 100 = 97$ である。このハッシュ値97は登録部14に入力される。

【0065】次に、登録部14は、この変換されたハッシュ値97と第2索引テーブル30に格納されているハッシュ値99との大小を比較し $97 \leq 99$ と認識する。同様にハッシュ値97とハッシュ値91との大小を比較し、 $91 < 97$ と認識する。これにより、登録部14は、ハッシュ値97を格納するハッシュ値群の所定範囲は、 $91 < 97 \leq 99$ と認識する。このとき、登録部14に格納されているハッシュ値群の先頭の値99より、登録部14は、ハッシュ値99が先頭に格納されている第1ブロック41をハッシュ値97が格納されるブロックと特定する。

【0066】次に、第2索引テーブル30に格納されたハッシュ値99に対応するポイントの指示に従い、第1ブロック41において変換したハッシュ値97と格納してあるハッシュ値とを先頭のハッシュ値99から順に対比することにより同一のハッシュ値97を検索する。

【0067】このとき、第1ブロック41には同一のハッシュ値97は格納されていないものとする。登録部14は、未だハッシュ値97が格納されていないことを認識すると、第1ブロック41にハッシュ値97を登録する。また、このハッシュ値97を変換する元になった従業員番号5443とデータ部50における従業員番号5443の格納位置を示すポイントとを登録する。また、ハッシュ値97下に属する従業員番号の格納数を1つ換算する。

【0068】なお、第1ブロック41に既にハッシュ値97が格納されている場合は、同一のハッシュ値97を検索すると、登録部14は、ハッシュ値97に対応する従業員番号を格納するセルに既に格納されている従業員

番号5443に加え、従業員番号4354を登録する。また、従業員番号4354のデータ部50における格納位置を示すポイントを登録する。さらに、ハッシュ値97下に属する従業員番号の格納数1に1つ加算し2とする。

【0069】このようにして、次々とデータを登録すると、図1に示すデータ部50と第1索引テーブル40とになる。このようなデータの格納状況において、データ部50から、例えば、氏名佐々木、従業員番号4354を検索する場合のデータ管理装置10の動作を図1と図3とを用いて以下に説明する。

【0070】まず、検索部13に従業員番号4354が入力される。すると、検索部13はハッシュ関数を呼び出し、 $h = (43 + 54) \bmod 100 = 97$ の演算を行い、従業員番号4354をハッシュ値97に変換する。次に、検索部13はハッシュ値97により、第2索引テーブル30において、変換されたハッシュ値97と格納されているハッシュ値99とを比較し、 $97 \leq 99$ と認識する。

【0071】同様にハッシュ値97とハッシュ値91との大小を比較し、 $91 < 97$ と認識する。これにより、検索部13は、ハッシュ値97を格納するハッシュ値群の所定範囲は、 $91 < 97 \leq 99$ と認識する。このとき、検索部13に格納されているハッシュ値群の先頭の値99より、検索部13は、ハッシュ値99が先頭に格納されている第1ブロック41をハッシュ値97が格納されるブロックと特定する。

【0072】次に、第2索引テーブル30に格納されたハッシュ値99に対応して格納されたポイントの指示に従い、第1ブロック41において変換したハッシュ値97と格納してあるハッシュ値とを先頭のハッシュ値99から順に対比することにより同一のハッシュ値97を検索する。

【0073】この同一のハッシュ値97を検索すると、次に検索部13は、このハッシュ値97下に属する従業員番号の格納数2を検索し、格納されている従業員番号と、検索用に入力された従業員番号4354との比較を最大2回行えば良いことを認識する。そして、従業員番号を上から順に対比し、2回目の対比において、同一の従業員番号4354を検索し、この同一の従業員番号4354を検索対象として特定する。

【0074】最後に、この従業員番号4354に対応して格納されたデータ部50における従業員番号4354の格納位置を示すポイントに従い、データ部50から氏名佐々木を検索する。

【0075】このとき、ハッシュ値97下に属する従業員番号の格納数が1である場合は、従業員番号の対比を一度だけ行い検索対象を特定し、従業員番号4354に対応するポイントに従ってデータ部50から氏名佐々木を検索する。

10

20

30

40

50

【0076】なお、検索部13は、第1索引テーブル40においてハッシュ値97を検索できないときは該当氏名なしのメッセージを出し、CRT等の出力装置が表示するようにされている。

【0077】〈データの削除〉次に、データ部50に格納した氏名山田、従業員番号8110を削除する場合のデータ管理装置10の動作を図1と図4とを用いて以下に説明する。

【0078】まず、削除部15に氏名山田と従業員番号8110とが入力されると、ハッシュ変換部12に従業員番号8110が入力される。すると、ハッシュ変換部12がハッシュ関数を呼び出し、 $h = (81 + 10) \bmod 100 = 91$ の演算を行い、従業員番号8110をハッシュ値91に変換する。このハッシュ値91は削除部15に送られる。

【0079】次に、削除部15は、この変換されたハッシュ値91と第2索引テーブル30に格納されているハッシュ値99との大小を比較し $91 \leq 99$ と認識する。次にハッシュ値91とハッシュ値91とを比較し、 $91 \leq 91$ と認識する。この場合は、変換したハッシュ値91がハッシュ値群の先頭の値91と一致するため、削除部15は、ハッシュ値91が先頭に格納されている第2ブロック42をハッシュ値91が格納されるブロックと特定する。

【0080】次に、第2索引テーブル30に格納されたハッシュ値91に対応するポインタの指示に従い、第1ブロック41において変換したハッシュ値91と格納してあるハッシュ値とを先頭のハッシュ値99から順に対比することにより同一のハッシュ値91を検索する。

【0081】このとき、ハッシュ値91下に属する従業員番号の格納数が1である場合は、入力された従業員番号8110と格納されている従業員番号8110とを1回対比し、削除データに対応する従業員番号8110を特定する。そして、第1索引テーブル40からハッシュ値91と従業員番号8110とポインタとを削除する。また、ハッシュ値91下に属する従業員番号の格納数を1つ減算する。

【0082】そして、削除するポインタをもとに、データ部50から氏名山田と従業員番号8110とを削除する。一方、従業員番号の格納数が複数の場合、すなわち、図1のように、従業員番号8110の他に従業員番号1081が格納されている場合は、入力された従業員番号8110と格納されている従業員番号との対比を最大2回行えば良いことを認識する。そして、従業員番号の格納順に上から対比を行い、一回目の対比で従業員番号同士が一致するため、この従業員番号8110を削除氏名に対応する従業員番号8110と特定する。そして、第1索引テーブル40の第2ブロック42から従業員番号8110とこの8110に対応するポインタとを削除する。また、ハッシュ値91下に属する従業員番号

の格納数を一つ減算する。

【0083】そして、ポインタをもとに、データ部50から氏名山田と従業員番号8110とを削除する。

〈データの更新〉データ部50に格納した氏名を更新する場合におけるデータ管理装置の動作は、データ部50から氏名と従業員番号とを削除し、続けて氏名と従業員番号とを追加する場合の動作と同様にして行われる。したがって、詳細な説明は省略する。

【0084】〈実施の形態の効果〉以上説明した実施の形態にかかるデータ管理装置の効果を以下に述べる。データ管理装置10によると、第2索引テーブル30と第1索引テーブル40とには、ハッシュ変換手段12により変換されたハッシュ値が格納されている。このハッシュ値は、元の識別情報である従業員番号に比べると2分の1のビット長である。従って、第2索引テーブル30と第1索引テーブル40とに格納された情報量は全体として減少する。このため、検索部13は、データ検索を短時間で行うことができる。

【0085】また、第2索引テーブル30を有しているため、変換されたハッシュ値が格納されたブロック41、42を早く検索することができる。また、登録部14と削除部15とを備えることにより、常に最新のデータ部50の格納内容に対応する索引テーブル30と第1索引テーブル40とを再編成することができる。この再編成も短時間で処理することができる。

【0086】また、第1索引テーブル40に従業員番号を格納するため、ハッシュ値の衝突が生じる場合であっても、検索手段は正確にデータを検索することができる。

【0087】

【発明の効果】本発明のデータ管理装置によると、ハッシュ変換手段によりデータの識別情報をハッシュ関数に従って識別情報のビット長よりも短いハッシュ値に変換するため、索引テーブルに格納する情報量の減少を図ることができる。このため、記憶手段に格納したデータの検索時間を短縮することができる。また、ハッシュ値の衝突の発生に拘わらず適切にデータを検索することができる。

【図面の簡単な説明】

【図1】図1は、本発明の実施の形態にかかるデータ管理装置の概要を示すブロック図である。

【図2】図2は、データ部にデータを登録する際におけるデータ管理装置の動作の手順を示すフローチャートである。

【図3】図4は、データ部に格納したデータを検索する場合におけるデータ管理装置の動作の手順を示すフローチャートである。

【図4】図4は、データ部からデータを削除する際におけるデータ管理装置の動作の手順を示すフローチャートである。

【図5】図5は、従来のデータ管理装置の概要を示すブロック図である。

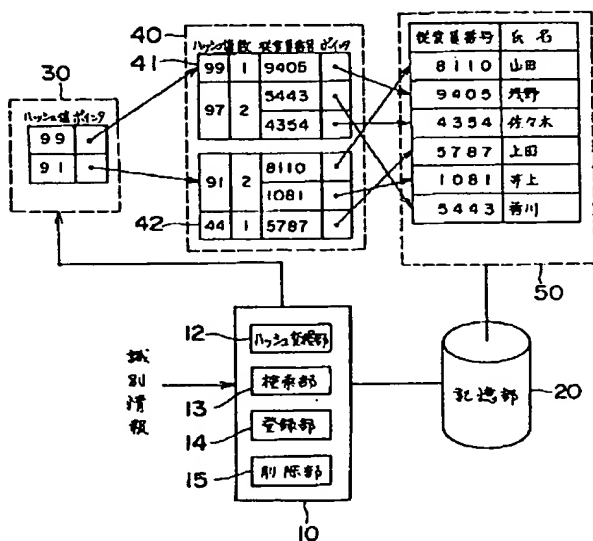
【符号の説明】

- 12 ハッシュ変換部
- 13 検索部
- 14 登録部
- 15 削除部

- 20 記憶部
- 30 第2索引テーブル
- 40 第1索引テーブル
- 41 第1ブロック
- 42 第2ブロック
- 50 データ部

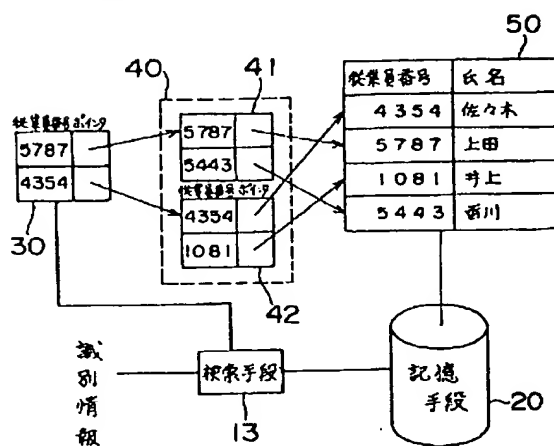
【図1】

本発明の実施の形態にかかるデータ管理装置の概要を示すブロック図



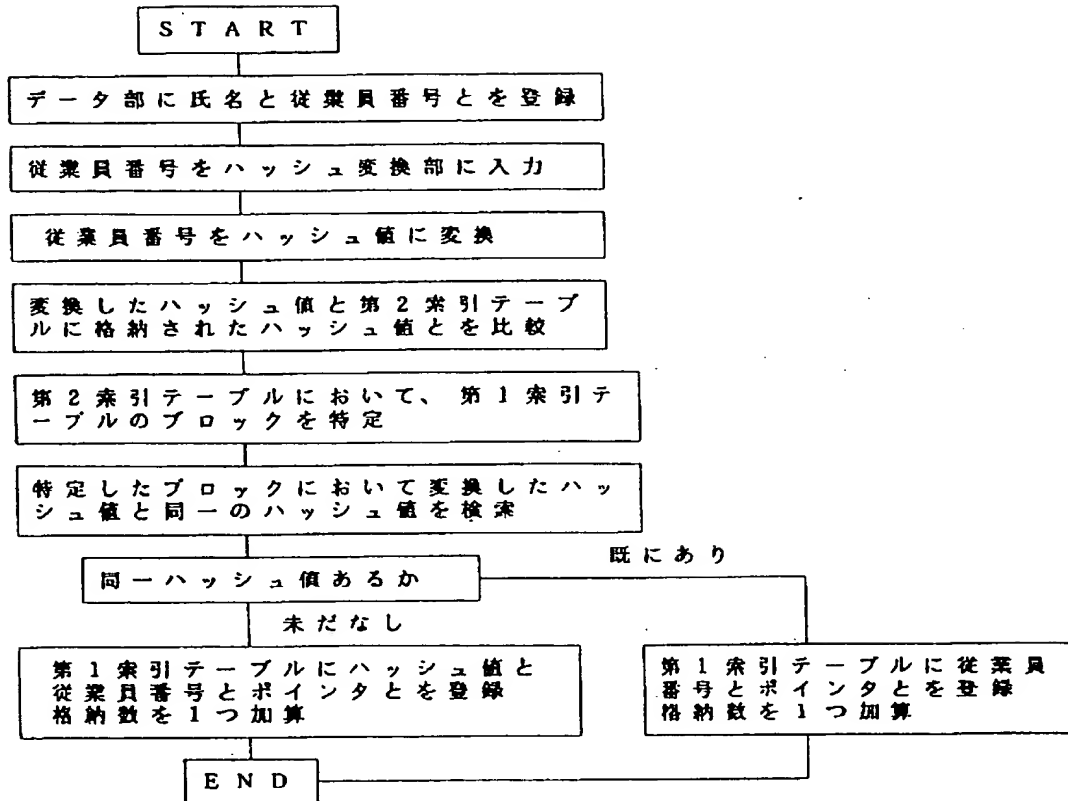
【図5】

従来のデータ管理装置の概要を示すブロック図



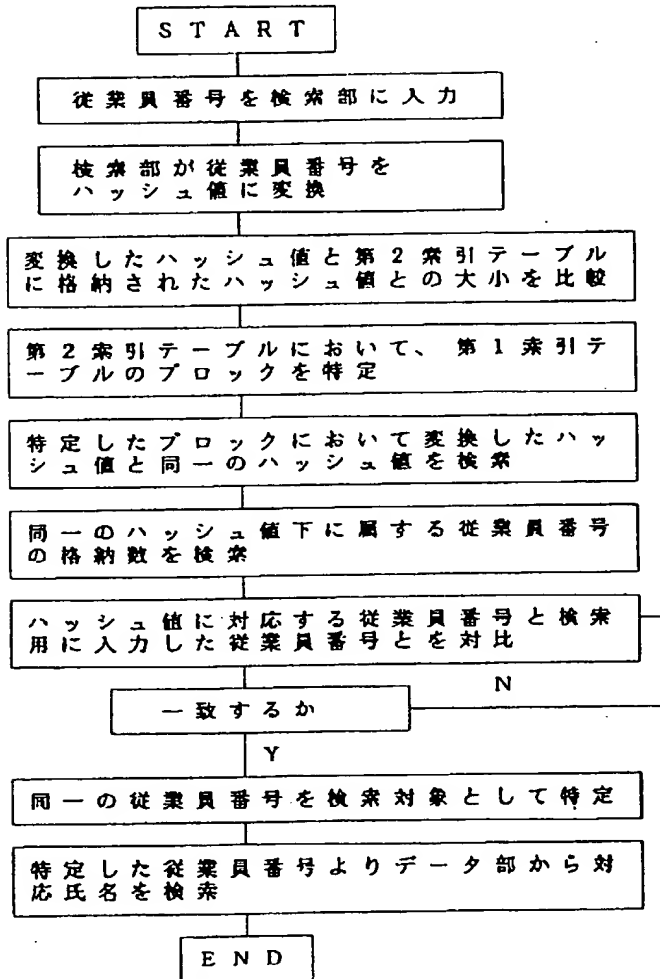
【図2】

データ部にデータを登録する際におけるデータ管理装置の動作の手順を示すフローチャート



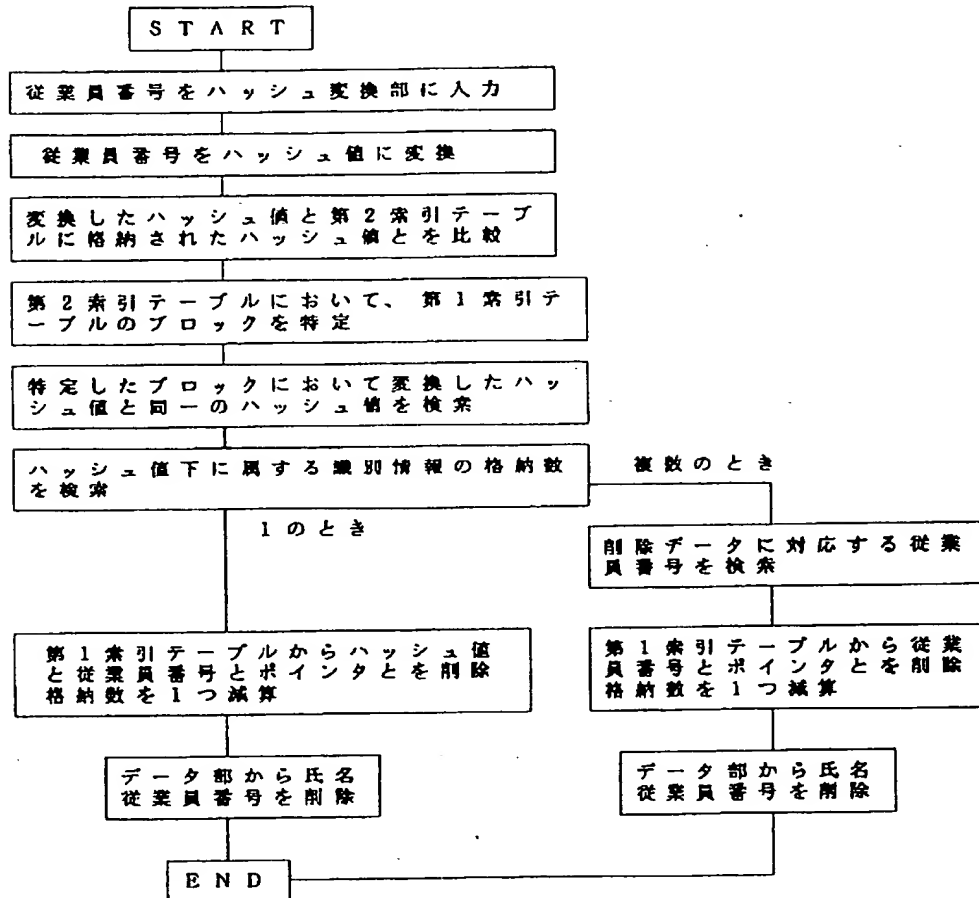
【図3】

データ部に格納したデータを検索する際におけるデータ管理装置の動作の手順を示すフローチャート



【図4】

データ部からデータを削除する際におけるデータ管理装置の動作の手順を示すフローチャート



フロントページの続き

(72)発明者 守屋 圭司

長野県松本市深志1丁目765番地 株式会
社富士通パソコンラボ内